

Data Management and Transport

Arie Shoshani

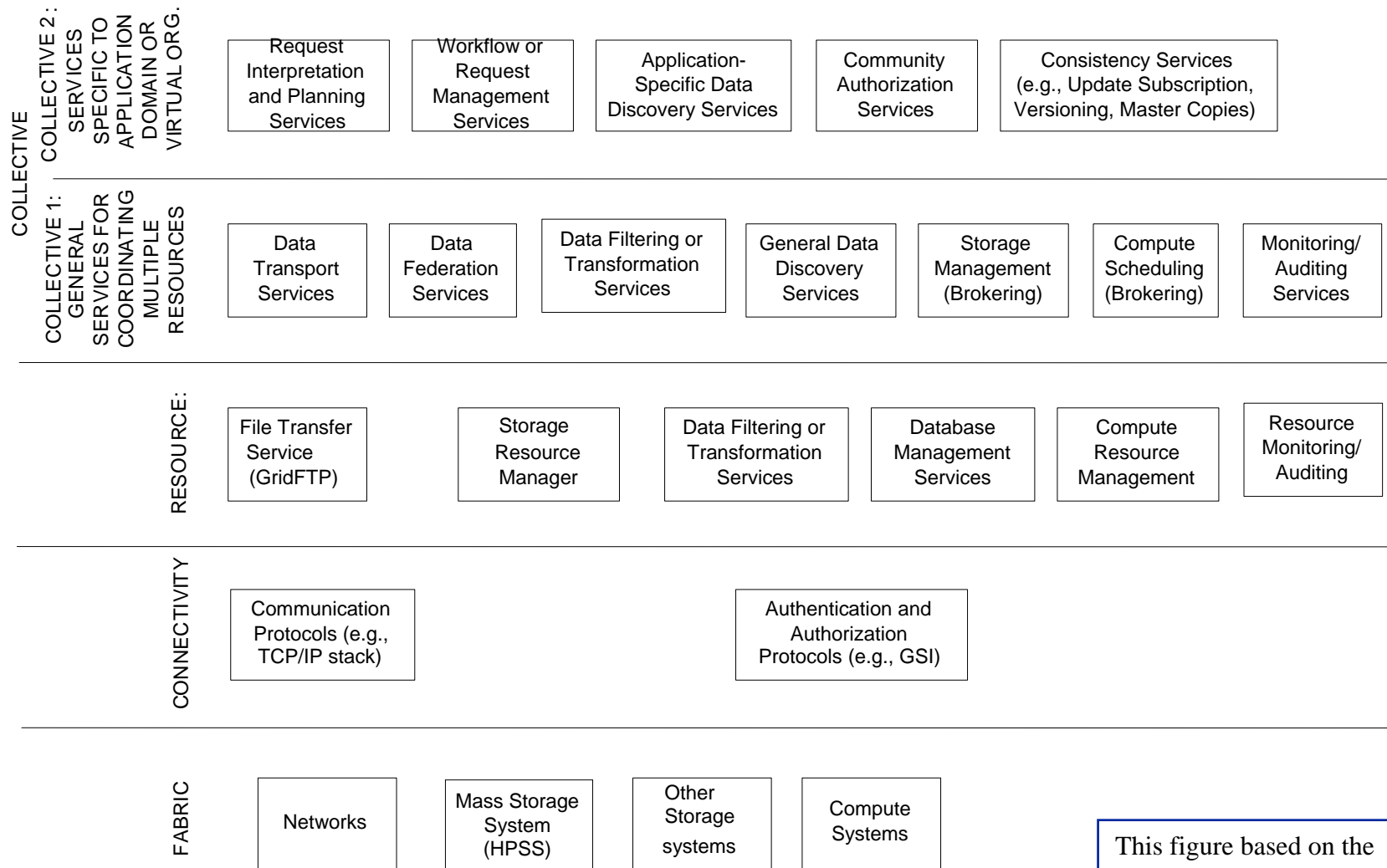
Scientific Data Management Group
Lawrence Berkeley National Laboratory

National Collaboratory Program Meeting,
August, 2004

- **Data**
 - data items organized in containers
 - e.g. Files, Blobs
 - Data-objects: are referenced by container names
- **Data Management =**
 - Storage management +
 - Data-object management
- **Data transport =**
 - Space management +
 - Data-object movement

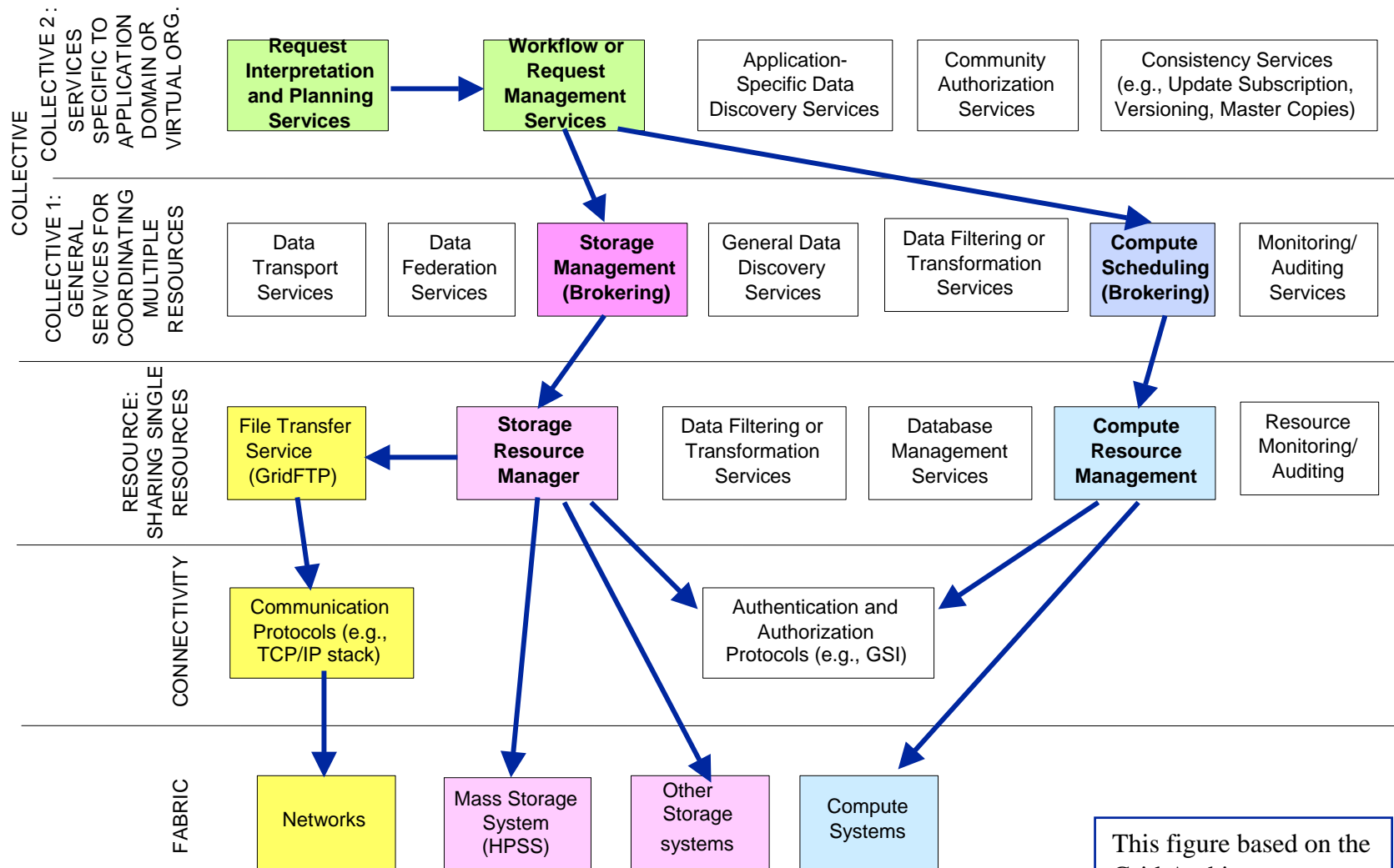
Grid = Distributed Computing?

- Isn't it just "distributed computing"? Yes, but ...
- The Grid is a **common infrastructure of services** that **enable** distributed computing
- Suppose you want to run a large compute/data-intensive job (need to use community resources)
 - Need to get compute resource
 - Need to get storage resources
 - Need to co-locate data objects and compute modules
 - Need to schedule job
 - May need to set up parallel sub-jobs, maybe to set up data streaming => workflow support
 - Need to run workflow (job)
 - Need to move resulting data objects to other storage resources (e.g. an archive or user site)
 - Need to monitor errors and failures
 - Need to recover from transient errors
 - Need to enforce security
- **Need a single infrastructure and an architecture for the services**



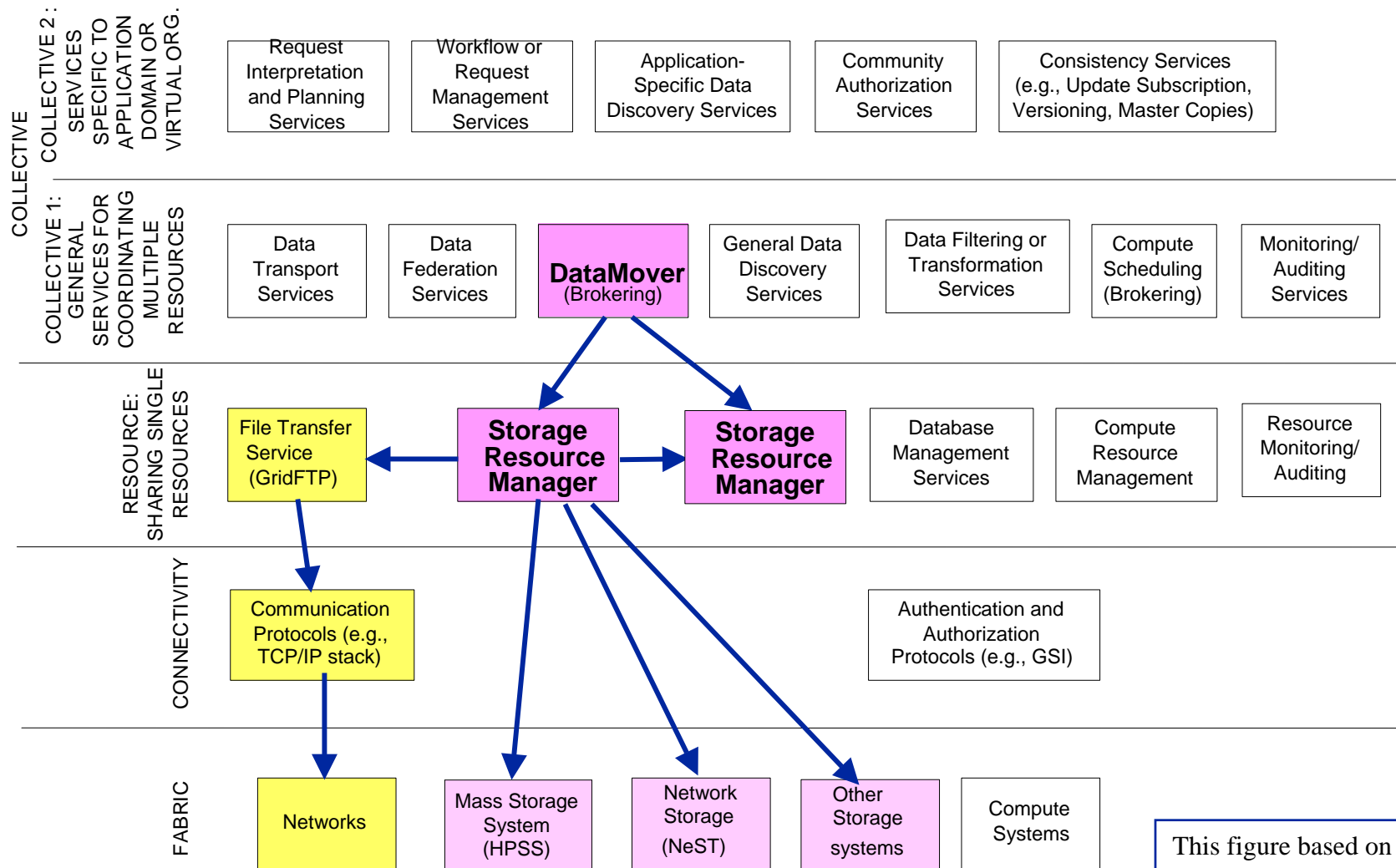
This figure based on the Grid Architecture paper by Globus Team

Grid - The Vision



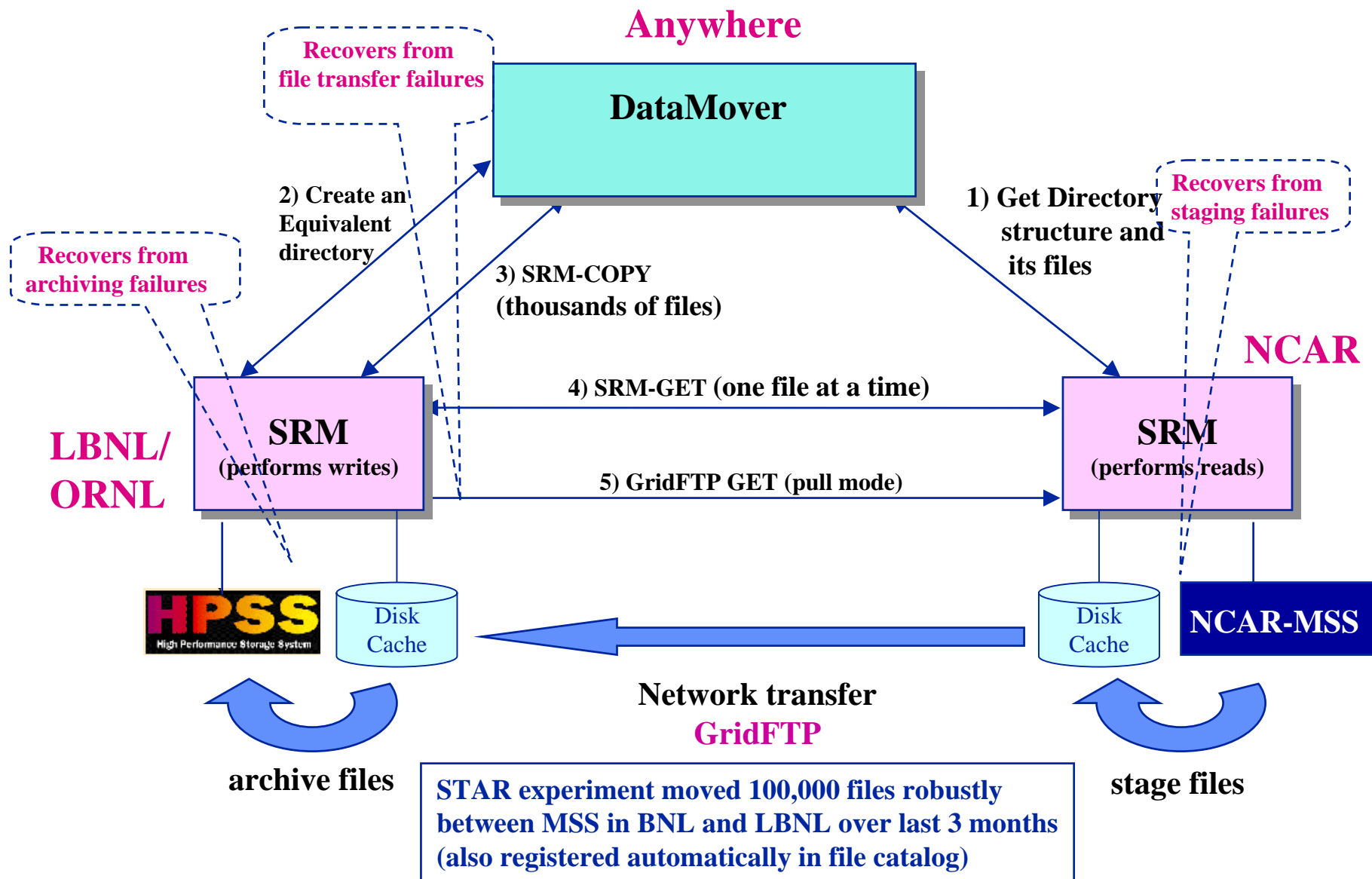
This figure based on the Grid Architecture paper by Globus Team

The DataMover is a Brokering Service That Invokes SRMs and Transfer Services

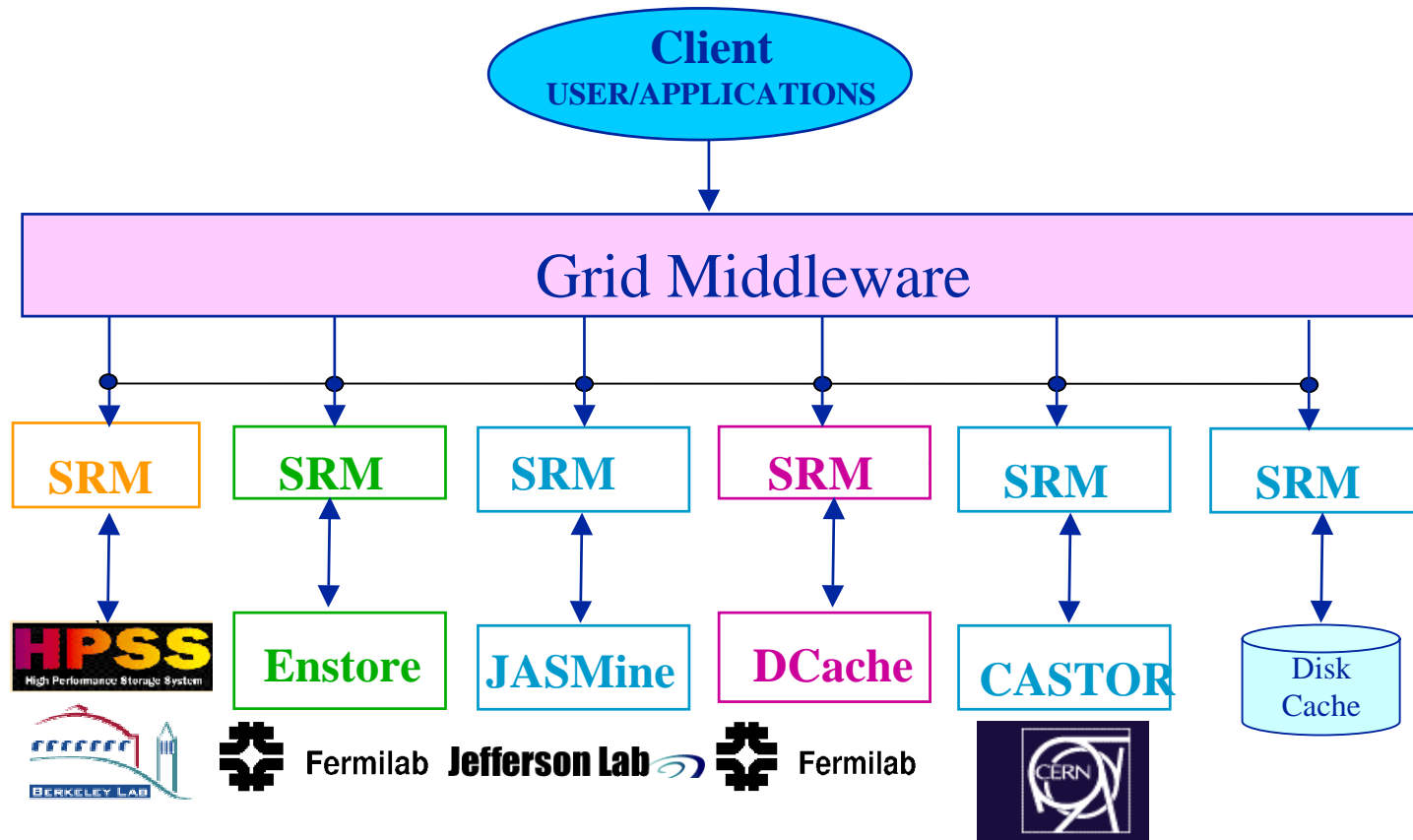


This figure based on the
Grid Architecture paper
by Globus Team

DataMover: SRMs used in ESG for Robust Multi-file replication



Uniformity of Interface → Compatibility of SRMs



What did we learn?

- **Standardization and Composition of Services Works!**
 - The Grid is NOT about standardizing middleware software
 - It is about a framework for standard functional specifications of services that inter-operate
 - The framework of a layered architecture works: services can be composed from other peer or lower level services
 - Want multiple Grid services providers that target common functional interfaces
- **Analogy**
 - Structured Query Language (SQL) for relational database systems
 - Multiple implementations, some commercial, some open source
 - Compete based on features, robustness, recovery, ...
- **Cost**
 - Getting technology to production is expensive – often not directly funded
 - Important to do, that's how to measure success
 - Need to support explicitly activity of adapting technology (Bridge the gap between users to technologist)



Data Management and the Grid: Heaven or Hell? (or what's still missing)



- Potentially heaven, but ... still need to go through hell to get there
- What's hellish? – data management perspective
- Space management issues
 - Large-scale robust data transport coupled with space management
 - The space reservation dilemma - incremental allocation?
 - Co-scheduling of compute and storage resources
 - Identify bottlenecks, automatic replication
 - Automated space management and garbage collection
 - Space and data objects lifetime mechanisms
 - Ensure that important data-objects are not lost



Data Management and the Grid: Heaven or Hell? (or what's still missing)



- **What's hellish? – data management perspective**
- **Allocation and authorization issues**
 - **Space management and allocation**
 - Managed by virtual organizations
 - Allocating quotas, enforcing and reporting resource usage
 - **Authorization management and enforcement**
 - Data-objects authorization
 - Centralized? How to coordinate authorization updates?
- **Performance prediction**
 - It's not only the space availability and the network speed
 - It's also the I/O allocation at the storage system
 - And also estimating access from MSS (it is on tape/disk?)

General “hell” that affects data management too

- **Distributed system error detection and recovery**
 - Tracking intra and inter component failures
- **Running very long jobs (transactions)**
 - Tracking status
 - Suspend/resume operations
- **Security**
 - Many models: GSI, Kerberos, SSL, one-time-password
 - Firewalls
- **Adapting to changing technology**
 - Things break as new versions introduced

- **Good reasons to be optimistic**
 - Lots of example success stories – even production deployment
- **For successful collaboratories we need to:**
 - Standardize on functional specification of services
 - Allow multiple implementations – can't dictate software uniformity
 - Develop robust interoperating products – need coordinated infrastructure for testing and distribution
- **Need future development in areas of:**
 - Dynamic storage management
 - Dynamic data object management
 - Storage resource allocation and authorization enforcement
 - Data object authorization enforcement
 - Failure tracking and reporting
 - Performance estimation